

REMARKS

Claims 1-24 are currently pending in the subject patent application.

In the above-identified Office Action, made final, claims 1-4, 6-12, 14-20, and 22-24 were rejected under 35 U.S.C. 103(a) as being unpatentable over Rezaul Islam et al. (U.S. Patent No. 6,282,670) in view of Burns et al. (U.S. Patent No. 6,088,694) and further in view of Hesselink et al. (U.S. Patent Publication No. 2005/0149481), since the Examiner stated that as per claim 1, Rezaul Islam et al. discloses a method for recovering data in a redundant data storage system having a plurality of data storage units, said method comprising: storing said data on said plurality of data storage units according to a redundant data storage method [Col. 2, lines 18-29]; removing one of said plurality of data storage units [Col. 4, lines 33-42]; while said one of said plurality of data storage units is removed, changing a portion of said data on the remainder of said plurality of data storage units and [Col. 4, lines 36-43]; replacing said one of said plurality of data storage units [Col. 1, line 66 to Col. 2, line 6]; and updating said one of said plurality of data storage units [Col.2, lines 18-27]. The Examiner continued that Rezaul Islam et al. does not specifically teach storing a record of said changes in a delta file and updating those portions of data recorded in said delta file as required by the claim.

The Examiner then stated that Burns et al. discloses storing a record of changes in a delta file and updating portions of data recorded in the delta file [Col. 5, line 61 to Col. 6, line 5] in order to achieve efficient and cost effective backup of data [Col. 6, lines 3-5], and that since the technology for implementing a storage recovery system with storing a record of changes in a delta file and updating portions of data recorded in the delta file was well known as evidenced by Burns et al., an artisan would have been motivated to implement this feature in the system of Rezaul in order to achieve efficient and cost-effective backup of data. The Examiner concluded that it would have been obvious to one of ordinary skill in the art at the time of invention by applicants to modify the system of Rezaul Islam et al. to include storing a record of changes in a delta file and updating portions of data recorded in the delta file because this would have achieved efficient and cost effective backup of data [Col. 6, lines 3-5] as taught by Burns et al.

The Examiner continued that Rezaul Islam et al. and Burns et al. do not specifically teach starting a delta log concurrently with said step of removing one of said plurality of data storage units as required, but that Hesselink et al. discloses starting a delta log concurrently with said step of removing one of said plurality of data storage units [paragraph 166, ll. 9-20] to allow the storage unit to later obtain changes that were made during the offline period [paragraph 166, ll. 15-19]. The Examiner then stated that since the technology for implementing a storage recovery system with starting a delta log concurrently with said step of removing one of said plurality of data-storage units was well known as evidenced by Hesselink et al., an artisan would have been motivated to implement this feature in the system of Rezaul and Burns since this would have allowed the storage unit to later obtain changes that were made during the offline period. The Examiner concluded that it would have been obvious to one of ordinary skill in the art at the time of invention by applicants to modify the system of Rezaul Islam et al. and Burns et al. to include starting a delta log concurrently with said step of removing one of said plurality of data storage units because this would have allowed the storage unit to later obtain changes that were made during the offline period [paragraph 166, ll. 15-19] as taught by Hesselink et al.

Applicants respectfully disagree with the Examiner concerning the rejection of claims 1-4, 6-12, 14-20, and 22-24 under 35 U.S.C. 103(a) as being unpatentable over Rezaul Islam et al. in view of Burns et al. and further in view of Hesselink et al. for the reasons to be set forth hereinbelow.

The Examiner rejected claims 9 and 17, using the same rationale as used in the rejection of claim 1. The Examiner then stated that Rezaul Islam et al. further discloses a redundant data storage system capable of fast restoration of serviced data storage units comprising: a plurality of data storage units [Col. 4, lines 33-44]; and a controller that stores data on said plurality of data storage units according to a redundant data storage method, changes a portion of said data after taking one of said plurality of said data storage units off line, stores a record of the changes in a delta log that are made to the remainder of the plurality of said data storage units, brings said one of said plurality of said data storage units online, and updates said one of said plurality of said data storage units by updating those portions of data recorded in said delta file [Fig.1,

controller 120; and Col. 17, lines 46-60].

Applicants respectfully disagree with the Examiner concerning the rejection of claims 9 and 17 under 35 U.S.C. 103(a) as being unpatentable over Rezaul Islam et al. in view of Burns et al. and further in view of Hesselink et al. for the reasons to be set forth hereinbelow.

Dependent claims 2-4, 6-8, 10-12, 14-16, 18-20, and 22-24 were rejected under 35 U.S.C. 103(a) as being unpatentable over Rezaul Islam et al. (U.S. Patent No. 6,282,670) in view of Burns et al. (U.S. Patent No. 6,088,694) and further in view of Hesselink et al. (U.S. Patent Publication No. 2005/0149481). Since dependent claims 2-4 and 6-8 depend from subject independent claim 1, dependent claims 10-12 and 14-16 depend from subject independent claim 9, and dependent claims 18-20 and 22-24 depend from subject independent claim 17, and applicants believe that claims 1, 9 and 17 are patentable over these references for the reasons to be set forth hereinbelow, applicants believe that no further response is necessary concerning the rejection of dependent claims 2-4, 6-8, 10-12, 14-16, 18-20, and 22-24.

Claims 5, 13, and 21 were rejected under 35 U.S.C. 103(a) as being unpatentable over Rezaul Islam et al. (U.S. Patent No. 6,282,670) in view of Burns et al. (U.S. Patent No. 6,088,694), Hesselink et al. (U.S. Patent Publication No. 2005/0149481) as applied to claims 1, 9, and 17 respectively, and further in view of McCabe et al. (U.S. Patent Publication No. 2002/0016827), since as per claims 5, 13 and 21, the Examiner stated that McCabe et al. discloses the redundant data storage method comprises remotely mirroring the data [Fig. 3; paragraph 20] in order to provide better fault tolerance and/or disaster recover [paragraph 2]. The Examiner continued that since the technology for implementing a storage recovery system with remote mirroring was well known as evidence by McCabe et al., an artisan would be have been motivated to implement this feature in the system of Rezaul Islam et al., Burns et al. and Hesselink et al. in order to provide better fault tolerance and/or disaster recovery. The Examiner concluded that it would have been obvious to one of ordinary skill in the art at the time of invention by applicants to modify the system of Rezaul Islam et al., Burns et al. and Hesselink et al. to include remote mirroring because this would have provided better fault tolerance and/or disaster recovery (paragraph 2) as taught by McCabe et al.

Applicants respectfully disagree with the Examiner concerning the rejection of claims 5, 13, and 21 under 35 U.S.C. 103(a) as being unpatentable over Rezaul Islam et al. in view of Burns et al., Hesselink et al. as applied to claims 1, 9, and 17 respectively, and further in view of McCabe et al. for the reasons to be set forth hereinbelow.

Reconsideration and reexamination are respectfully requested.

Turning now to the rejection of claims 1-4, 6-12, 14-20, and 22-24 under 35 U.S.C. 103(a) as being unpatentable over Rezaul Islam et al. in view of Burns et al. and further in view of Hesselink et al., the Examiner stated that Hesselink et al. discloses starting a delta log concurrently with said step of removing one of said plurality of data storage units [Paragraph [166], ll. 9-20] to allow the storage unit to later obtain changes that were made during the offline period [Paragraph [166], ll. 15-19].

First, applicants wish to point out that the word “concurrent” means happening at the same time as something else and paragraph [166] of Hesselink et al. does not suggest or imply such a requirement. In lines 6-9 of paragraph [166] it is stated that: “For example, a user may take computer **72** on an airplane and continue to edit ‘my documents.doc’ even though she or he does not have any network connection available.” Generally, laptop computers are placed in the **off** condition while going to and from airports. Thus, there would be no reason for a delta log to be started concurrent with the removal of the laptop computer, nor does Hesselink state such an action.

Further, in paragraph [0078] of Hesselink et al., it is stated that: “The term ‘Delta’ refers to differential data that is the result of calculating differences between files that is performed for difference compression used in updating a file according to the present invention (e.g., the difference between the file as originally downloaded and the file as edited, or the difference between the currently edited file and the most recent previously edited file).” In paragraph [0166] of Hesselink et al. it is stated that: “Another situation that may give rise to file data version conflict is offline use. The present invention may allow user modules to continue to serve data access requests directly from their respective local caches even after the user modules are no longer connected to the network, or when network communication is not available. For

example, a user may take computer 72 on an airplane and continue to edit 'myDocument.doc' even though she or he does not have any network connection available. During the time the computer 72 is offline or disconnected from the network user module 72su may still be able to store changes that are made with respect to the file data associated with 'myDocument.doc' and/or file overhead information in storage device 18a, hence providing normal storage device functionalities to the local applications. When the network communications are resumed, user module 72a detects the online condition and begins contacting device module 74sd to submit the changes that were made during the offline period. While user module 72su was offline, however, device module 74sd or one or more other user modules may have updated the version of 'myDocument.doc'. As a result, a version conflict situation occurs because the Delta update submitted by user module 72su is not based on the current official file reference version in this situation. For this reason, device modules are configured to store only one version, of a file data to be the current file version.

All other versions are generated based on Deltas which define the revisions to the file.” (Emphasis added by applicants.). Thus, Hesselink et al. clearly does not teach the concurrent starting of a delta log with removal of the computer connection, as suggested by the Examiner. Rather, the Delta update is associated with the user module while the device module is configured to store **only one version** of a file data; that is, the device module is **not** changed during the removal of the storage unit as is required by subject claim 1. In fact, the device module does not “know” about the changes made until network communications are resumed and the changes have already been made on the user module. This means that there cannot be a starting of a delta log concurrently with said step of removing one of said plurality of data storage units which meets the limitations of subject claim 1, as suggested by the Examiner.

Subject claim 1 recites: “... changing a portion of said data on the remainder of said plurality of data storage units during the period when one of said plurality of data storage units is removed; storing a record of said changes in said delta log;” Clearly, the present claimed invention requires that the delta log store changes to the **rest** of the storage units, which changes are **transferred to the removed storage unit** in said step of: “... updating said one of said plurality of data storage units by updating

those portions of data recorded in said delta log.” Hesselink et al., by contrast, requires that this not be done, since it would cause a version conflict situation.

Applicants therefore, fail to understand in what manner the Examiner is using the Hesselink et al. reference in combination with Rezaul Islam et al. and Burns et al.

Returning to the Examiner’s conclusion that it would have been obvious to one of ordinary skill in the art at the time of invention by applicants to modify the system of Rezaul Islam et al. to include storing a record of changes in a delta file and updating portions of data recorded in the delta file because this would have achieved efficient and cost effective backup of data [Col. 6, lines 3-5] as taught by Burns et al., applicants wish to point out that the defective drives of Rezaul Islam et al. are taken off line since they are defective. The files of Burns et al. being updated are not defective drives; therefore, applicants fail to understand how Burns et al. cures a problem for Rezaul Islam et al., since a user would not wish to access a defective drive.

Applicants wish to direct the Examiner’s attention to Col. 4, lines 21-46, of Burns et al. wherein it is stated that “The present invention provides a computer system that permits continuous availability of a file that is stored in a file management system and linked or referenced by a database management system (DBMS) through a DATALINK data type, even while the file is being modified at the file management system with either the append or update operation. Applications gain access to a linked file name through the DBMS, access file data through local file systems, and make modifications to that data. During this modification, other applications or users can access the immediately previous version of the file which is registered in the DBMS. When a modifying application finishes writing file data on the local file system, it closes its write access and transactionally updates the Datalink reference to the linked file, to reference the changed file data.” In Col 5, line 25, to Col. 6, line 5, Burns et al. continue that: “The invention can be applied to the operation of updating a linked file A1, which was described above. In accordance with the invention, the procedure could be changed as follows: (a) Make a copy of file A1 using the filesystem services. Let the copy of the file be called A2. (b) Modify A2. Note that file A1 remains linked while the user modifies A2 and is available for the DBMS queries. (c) Make new metadata based on the content of A2. (d) Update the row which referenced file A1 in the database with the metadata and

reference to A2. The update processing in the DBMS involves the following. File A1 is “unlinked” and A2 is “linked” as part of the same transaction. Furthermore, as part of link processing of A2, backup for file A2 is initiated. To provide database consistency, integrity and recoverability guarantees, the linked version of the file must be able to be recalled from a backup server. Consider that the file system fails and the contents of a linked file are lost. The DATALINK value in the database is now inconsistent and the file contents, as they were at link time must be restored. Alternatively, consider that the database experiences system failure and has to be recovered to a previous image of the database. A Datalink column in the previous versions may link what is now an out of date file. To make the restored database consistent, the Data Linker must restore the old version of the file. For DBMSs that link and unlink files frequently, a large number of file backups will be performed by the DataLinker. For each backup, the number of bytes in the linked file will be sent to the backup server. This cost may become prohibitive. Consequently, there is also a need for an efficient backup mechanism to reduce the amount of information sent to a backup server in association with keeping file data and Datalink data type columns consistent and recoverable. The efficient backup involves that only the modified portions of a “versioned” file be backed up. In our example above, the reference file A1 is referred to as a “versioned” file. Assume file A1 is backed up in total. However, the backup operation of the new version of file A2 would involve backing up only the modified portions with respect to A1. The file changes from a prior version define what is called a delta file. This delta file compactly represents A2 as a set of changes to A1. By backing up this delta file instead of backing up the whole file A2, a computer system may reduce the cost of backup for frequently versioned data files.” (emphasis added by applicants).

Applicants wish to point out that the present claimed invention does not create a delta file by comparing two existing, already stored and linked, files (Please see underlined portion from Burns et al. immediately above.) and identifying the modified portions for a backed up file, as taught by Burns et al. Rather, the delta file of the present invention is recited as: “... storing a record of said changes in said delta log;” (Please see claim 1, hereof.). The changes referred to in claim 1 relate to: “... changing a portion of said data on the remainder of said plurality of data storage units during the

period when one of said plurality of data storage units is removed; ... and updating said one of said plurality of data storage units by updating those portions of data recorded in said delta log.”

Further, Burns et al. requires that A1 remain linked while the user modifies A2, A1 being available for the DBMS queries, while Rezaul Islam et al., by contrast, requires that the system has means for identifying a failed storage device and for **removing** the failed storage device from the RAID system. Rezaul Islam et al. in Col. 11, line 55 to Col. 12, line 20 states that: “Media errors on a physical drive can occur that result in the device not being able to supply the requested data for a stripe unit. If a media error occurs during a logical drive rebuild, then the data on the stripe cannot be reconstructed. For example, if a media error occurred while reading a strip unit on a first disk, then the corresponding (from the same stripe) stripe unit from the second disk could not be reconstructed. If this error is ignored, the next time the user accesses data in the stripe unit of the second disk, incorrect data will be supplied since it was not reconstructed correctly. To avoid this problem, some RAID controllers report the device with the media errors as a dead device. When this occurs, the entire logical drive will go off line since the data can no longer be reconstructed (at this point there are 2 stripe units with unknown data). When a logical drive is off line, the user cannot access any data on that logical drive and all data on that logical drive is considered to have been lost. To bring the logical drive back online, the user must replace the device that has the media error, and restore the lost data. To overcome this problem, this invention uses a table located in the adapter’s NVRAM to keep track of stripes that could not be reconstructed successfully. If a stripe could not be reconstructed successfully during a logical drive rebuild operation, an entry is made in a “bad stripe table” containing the logical drive and the stripe number that could not be rebuilt successfully. When the user tries to access data, this table is checked. If there is an entry in the bad stripe table for a stripe being accessed, the user will receive an error message. Although the user may lose a small portion of the data, the user is only presented with an error message instead of incorrect data.” (Emphasis added by applicants.).

Applicants fail to understand the Examiner’s conclusory statement that since the technology for implementing a storage recovery system with storing a record of changes

in a delta file and updating portions of data recorded in the delta file is well known as evidenced by Burns et al., an artisan would have been motivated to implement this feature in the system of Rezaul Islam et al. in order to achieve efficient and cost effective backup of data. Clearly, Rezaul Islam et al. does not require or teach this additional feature to achieve its goal of managing **defective** media in a RAID system. In fact, as stated hereinabove, the delta file of **Burns et al. compares the data from two existing and operating files**. Further, the teachings of Burns et al. cannot be used in Rezaul Islam et al., since at least some of the data may be lost in the logical drive of Rezaul Islam et al.

The Federal Circuit ruled in *In re Kahn* (Fed. Cir. No. 04-1616, March 22, 2006), that a Board of Patent Appeals and Interferences must articulate the motivation, suggestion or teaching that would have led the skilled artisan at the time of the invention to combine prior art elements to make the claimed invention. To establish a *prima facie* case of obviousness based on a combination of prior art elements, "the Board must articulate the basis on which it concludes that it would have been obvious to make the claimed invention, When the Board does not explain the motivation, or the suggestion or teaching, that would have led the skilled artisan at the time of the invention to the claimed invention as a whole, we infer that the Board used hindsight to conclude that the invention was obvious." Here, the Examiner has merely stated that since the technology for implementing a storage recovery system with storing a record of changes in a delta file and updating portions of data recorded in the delta file is well known as evidenced by Burns et al., an artisan would have been motivated to implement this feature in the system of Rezaul Islam et al. in order to achieve efficient and cost effective backup of data, and therefore, it would have been obvious to one of ordinary skill in the art at the time of invention by applicants to modify the system of Rezaul Islam et al. to include storing a record of changes in a delta file and updating portions of data recorded in the delta file because this would have achieved efficient and cost effective backup of data as taught by Burns et al. Applicants respectfully believe that this bare statement by the Examiner does not rise to a motivation, suggestion or teaching as required by the court in *In re Kahn*. Rather, applicants believe that the Examiner has used hindsight to construct the present claimed invention.

Section 2145 X. A. of the Manual Of Patent Examining Procedure states that: "[a]ny judgment on obviousness is in a sense necessarily a reconstruction based on hindsight reasoning, but so long as it takes into account only knowledge which was within the level of ordinary skill in the art at the time the invention was made and does not include knowledge gleaned only from applicant's disclosure, such a reconstruction is proper.' *In re McLaughlin* 443 F.2d 1392, 1395, 170 USPQ 209, 212 (CCPA 1971)." However, here the Examiner has combined references which cannot with benefit be combined, after applying the teachings of the present claimed invention.

Applicants therefore respectfully believe that the Examiner has failed to make a proper *prima facie* argument for obviousness as is required in a rejection under 35 U.S.C. 103(a), since there would be no motivation to combine Rezaul Islam et al. with Burns et al. Moreover, the combination of Rezaul et al. and Burns et al. with Hesselink et al. is improper since Hesselink et al. does not teach what the Examiner has stated it teaches. The further combination of these references with McCabe et al. does not justify this improper combination.

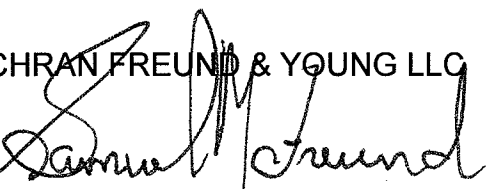
In view of the discussion presented hereinabove, applicants believe that subject claims 1-24 are in condition for allowance or appeal, the former action by the Examiner at an early date being earnestly solicited.

Reexamination and reconsideration are respectfully requested.

Respectfully submitted,

COCHRAN FREUND & YOUNG LLC

By:



Samuel M. Freund
Reg. No. 30,459/
2026 Caribou Drive, Suite 201
Fort Collins, Colorado 80525
Phone: (970) 492-1100
Fax: (970) 492-1101
Customer No.: 27479

Date: June 11, 2007